PRTS /786173
JC02 Rec'd PCT/PTO 2 8 FEB 2001

METHOD FOR ESTIMATING MOVEMENT BETWEEN TWO IMAGES

A

BACKGROUND OF THE INVENTION

The present invention concerns a method for estimating movement between two numerical images.

The movement between two successive images, I_1 and I_2, is generally defined in the form of a movement field associated with either of the images , I_1 and I_2 and constituted by movement vectors, each relating to one point of the image concerned. The movement vector is a two-dimensional vector representative of the difference of position between the pixel of the image I_1 and the associated pixel of the image I_2 relating to the same physical point of the filmed scene.

An evaluation of movement is useful in fields for processing the image requiring a knowledge of movements or disparities between two images. By way of examples, it is possible to cite the following spheres of applications :
- the compression of images : the evaluation method is used to limit the amount of data to code an image, the images being defined in relation to one another ;
- the compression of data in spaces of dimension greater than 2 ;
- video coding : the movement field defined from already coded images is then used to predict the next image ;
- medical imagery: the method for estimating the movement between two images is used to conduct an analysis of the movement of the heart for example ;
- remote monitoring : the method can be used to monitor road traffic ;
- three-dimensional reconstruction from multi-view images: the method is used to estimate disparities between various views.

So as to obtain this movement field, a method is known on how to break down the image into finished elements. These finished elements, which may for example be triangles or quadrangles, are determined by a meshwork whose nodes correspond to the tops of the finished elements. A movement vector is calculated for each node of the meshwork. Then, via the bias of an interpolation function, it is possible to deduce from this a movement vector for each point of the image in

question. The movement field is thus determined by a model of finished elements defining the meshwork used to partition the image into finished elements and the interpolation function making it possible to calculate the movement vector at any point of the image.

The meshwork used can be regular or irregular and needs to selected as sufficiently dense so as to model as best as possible the movement between the two images without however requiring an excessive quantity of calculation or data to be transmitted. This choice is made once only at the start of the method and this meshwork generally remains the same throughout the estimate.

The calculation of the movement vectors of the nodes of the meshwork can be carried out according to various methods. First of all there are putting into correspondence or "matching' methods consisting of testing a discrete set of possible values of movement vectors for each node of the meshwork and of retaining the best vectors according to a given criterion. A second method known as a transformed method consists of using the properties of the Fourier transform and its extensions so as to convert the movement into a phase jump in the transformed space. Finally, there is a third method known as a differential method for determining the movement vectors by optimising a mathematical criterion (for example a quadratic error between the image and its aforesaid value with the movement field). This method is most frequently used for estimating movement with modelisation by finished elements. A conventional differential method for optimising movement vectors is the Gauss-Newton method. The present application concerns more particularly the movement estimation method family using a model of finished elements and a differential method to determine the movement field.

Although widely used, this type of method does however have several drawbacks. The meshwork selected at the start of the method may prove to be inappropriate with respect to the semantic contents of the image, certain zones of the image requiring a denser meshing and others a more aerated meshing. In addition,

under the effect of the field of the movement vectors of the nodes of the meshwork, the initial meshing on the start image, for example I_2, is transformed into a new meshing on the other image, for example I_1. Then pathological situations may then occur at the level of the new meshing, such as :

- reversals of finished elements : finished elements reverse and cover others, thus destroying the property of partitioning of the range of the image any meshwork needs to verify,
- overflowing of the peripheral nodes of the moved meshwork after applying movement vectors beyond the range of the image I_1 : certain pixels of the image I_2 can be associated with pixels of the image I_1 situated outside the range of the image I_1. This is not strictly a problem, but it may be advantageous to force the peripheral nodes of the moved meshwork to remain inside the range of the image I_1.

OBJECTS AND SUMMARY OF THE INVENTION

One object of the invention concerns a method for estimating movement in which the meshwork is optimised during estimation so as to obtain at the end of the method a meshwork adapted to the semantic contents of the images. To this effect, the finished elements are refined during the movement estimate.

Another object of the invention is to improve the effectiveness of the Gauss Newton method so as to optimise the movement vectors of the nodes of the meshwork. To this effect, this optimisation is carried out on several resolution levels of the images.

Finally, another aim of the invention is to provide a method for estimating movement so as to avoid aforesaid pathological situations. To this effect, the invention provides adding during the movement vectors optimisation step constraints so as to avoid these situations.

Also, the invention concerns a method for estimating the movement between two numerical images I_1 and I_2 with luminance Y_1 and Y_2 and intended to generate for each point of coordinates x,y of the image I_2 a movement vector $\vec{d}(x,y)=(d_x,d_y)$ so

as to form an image $\hat{I}_2$ from the image I_1 with luminance $\hat{Y}_2(x,y)=Y_1(x-d_x,y-d_y)$ which is an approximation of the image I_2, characterised in that it comprises the following steps :

(a) defining an initial model of finished elements comprising a meshing whose nodes are points of the image I_2, a movement vector at each node of said meshing, and an interpolation formula for calculating the value of the movement vector of each point of the image I_2 from the values of the movement vectors of the nodes of the mesh to which it belongs,

(b) optimising the value of the movement vectors of the model according to a differential method,

(c) calculating a variation E between the image I_2 and the image $\hat{I}_2$ for each finished element or mesh,

(d) carrying out a finer meshing on a discrete fraction of the set of finished elements determined according to a criterion relating to the variations E and allocating a movement vector to each new meshing node,

(e) repeating the steps (b), (c) and (d) on the model of finished elements obtained at the end of the preceding step (d) until a stoppage criterion is satisfied.

According to an improved embodiment, for each numerical image I_1 and I_2 in addition a set of R images I_j^r is defined with a level of resolution r and luminance Y_i^r with r taking the values (0,…,R-1) and i the values 1 and 2, the images I_1^0 and I_2^0 corresponding to the numerical images I_1 and I_2, the steps (b) to (e) being conducted for each level of resolution r from the level r=R-1 to the level r=0.

Finally, according to a preferred embodiment, constraints are added to the movement of the finished elements at the time of optimising the movement vectors so as to avoid the reversal of the finished elements. According to another embodiment, it is also possible to introduce constraints so as to avoid the flowing over of the meshing obtained after applying the movement vectors beyond the sphere of the image I_1.

A

BRIEF DESCRIPTION OF THE DRAWINGS

Other characteristics and advantages of the invention shall appear on a reading of

the following detailed description with reference to the accompanying drawings on which :

- figure 1 represents a diagram of a first embodiment of the movement estimation method of the invention ;
- figure 2 shows the step (d) of the method of the invention, and
- figure 3 shows a diagram of an improved embodiment of the movement estimation method of the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A

Reference is first made to two numerical images I_1 and I_2 with respective luminance Y_1 and Y_2. The method of the invention consists of generating for each point P of coordinates (x, y) in the image I_2 a movement vector $\vec{d}(x,y)=(d_x,d_y)$. This vector is defined as being the vector able to construct from the image I_1 an image $\hat{I}_2$ with luminance $\hat{Y}_2$ $(x,y)=Y_1(x-d_x,y-d_y)$ which is an approximation of I_2. The movements are thus defined from the image I_1 towards I_2.

The sought-after movement field is defined by a model of finished elements. In the remainder of the description, the finished elements are regarded to be triangles without there being any limitation of the extent of the present application to this form of finished elements. As a result, the model of finished elements comprises a triangular meshing, movement vectors defined in the meshing nodes, and an interpolation formula for calculating the movement vector of the points inside the triangles.

The interpolation formula used to calculate the movement field at any point of the range of the image I_2 is the following :

If the point P of coordinates (x,y) is considered in the image I_2 belonging to the triangle e with vertices P_i, P_j and P_k with respective coordinates (x_i,y_i), (x_j,y_j) and (x_k,y_k), its movement vector is equal to

$$\vec{d}(x,y)=\sum_{l=i,j,k}\Psi_l^e(x,y).\vec{d}(x_l,y_l)$$

where Ψ_l^e represents a basic function associated with the triangle e

In the case of an affine interpolation, the $\Psi_l^e(x,y)$ represent the barycentric coordinates of the point P in the triangle e with vertices P_i, P_j, P_k. These functions are defined by the following equation :

$$\begin{cases} \begin{cases} \Psi_l^e(x,y) = \alpha_l + \beta_l x + \gamma_l y & (x,y) \in e \\ \sum_{l=i,j,k} \Psi_l^e(x,y) = 1 & \text{et} \quad \alpha_1, \beta_1, \gamma_1 \in \Re \end{cases} \\ \Psi_l^e(x,y) = 0 & (x,y) \notin e \end{cases}$$

namely $\Psi_i^e(x,y) = \dfrac{x_j y_k - x_k y_j + (y_j - y_k)x + (x_k - x_j)y}{x_j y_k - x_k y_j + x_k y_i - x_i y_k + x_i y_j - x_j y_i}$

The affine functions $\Psi_j^e(x,y)$ and $\Psi_k^e(x,y)$ deduced from the function $\Psi_i^e(x,y)$ by circularly permuting the indices i,j,k. It is also possible to use more evolved models of finished elements, the functions ψ then being able to be extended to polynomials with degree n=2, but the interpolation formula of the movement vectors then introduces first, second derivatives, etc. A miscellaneous choice of models of finished elements is shown in the work "Handbook of Numerical Analysis" ,by P.G. Ciarlet and J.L Lions, Volume 2, pages 59-99, published by North Holland.

According to the invention, as the movement is gradually estimated, the value of the movement vectors of the meshing nodes known as nodal vectors is optimised and the meshing is locally densified when this is necessary. Advantageously, this optimisation shall be carried out on several resolution levels starting with a low resolution level.

According to a first embodiment shown on figure 1, the method of the invention comprises five steps referenced (a) to (e).

According to step (a), an initial model of finished elements is defined by selecting points of the image I_2 according to an initial triangular meshing. The nodes of the meshing represent the vertices of triangles (finished elements) of the model. This meshing can be defined in any way, for example according to the needs of the application or prior knowledge or of the movement already calculated between two preceding images of the same video sequence. If no data concerning the initial meshing is specified, a regular staggered meshing is used. The meshes are then triangles. A nil value movement vector is then associated with each node of the meshing. The interpolation formula previously defined is also a data element of the initial model.

According to step (b), the value of the movement vectors of the model are optimised according to a differential method, such as the Gauss-Newton method or its Marquardt extension. This optimisation can be free, that is without constraints imposed on the possible values of the nodal vectors, or with constraints. The nodal vectors denote the movement vectors of the nodes of the meshing. Optimisation with constraints is directly linked to free optimisation and forms the subject of an embodiment given in detail later in the text.

The free optimisation technique used here makes use of the advantageous characteristics of the Gauss-Newton method (rapid convergence of the optimum) and the gradient method with adaptive step (global convergence towards a local optimum) to resolve the linear system to be followed. This technique is an iterative correction of the movement vectors $\vec{d}(x,y)$ making it possible to obtain from the start a rough approximation of the movement. The number of iterations k for this optimisation of the movement vectors is either specified by the user at the start of the method, or depends on a threshold linked to the maximum variation between two consecutive values of nodal vectors for two successive iterations. Below are details of the Marquardt extension of the Gauss Newton optimisation method.

The expression of the corrections δD^{k+1} at iteration k+1 of the movement vectors

according to this method is given by the following linear system :

$$D^{k+1} = D^{k}-[R^{k} + \alpha.I_{2N}]^{-1}.\nabla E^{k} \Leftrightarrow - H.\delta D^{k+1} = \nabla E^{k} \qquad (1)$$

with :

- D^{k+1} a column matrix of 2N elements including the components d_x and d_y of the nodal vectors on iteration k+1, N being the number of nodes of the meshing in the current step :
- D^{k} a column matrix of 2N elements including the components d_x and d_y of the nodal vectors on iteration k ;
- $H = [R^{k} + \alpha.I_{2N}]$
- I_{2N} the identity matrix with the dimension 2N ;
- $\nabla E^{k} = \begin{pmatrix} \nabla_x E^k \\ \nabla_y E^k \end{pmatrix}$ column matrix of 2N elements in which N elements $\nabla_{x,n}E^{k}$ and N elements $\nabla_{y,n}E^{k}$, n denoting a node of the meshing and taking in turn the values (1…N) ; with

$$\nabla_{x,n}E^{k} = 2.\sum_{e\in supp(n)}\sum_{(x,y)\in e} DFD_k(x,y).\frac{\partial I_1(x-d_x,y-d_y)}{\partial x}.\Psi_n^e(x,y)$$

$$\nabla_{y,n}E^{k} = 2.\sum_{e\in supp(n)}\sum_{(x,y)\in e} DFD_k(x,y).\frac{\partial I_1(x-d_x,y-d_y)}{\partial y}.\Psi_n^e(x,y)$$

where $DFD_k(x,y) = Y_2(x,y) - Y_1(x-d_x,y-d_y)$ on iteration k where sup p(n) represents the support of the basic function $\Psi_n^e(x,y)$ attached to the node n, that is all the triangles having the node n for a vertex ;

- $R^{k} = \begin{pmatrix} R^{k,xx} & R^{k,xy} \\ R^{k,yx} & R^{k,yy} \end{pmatrix}$ a square matrix with dimension 2N

where

$$R_{mn}^{k,xx} = 2.\sum_{e\in supp(mn)}\sum_{(x,y)\in e}\left(\frac{\partial I_1(x-d_x,y-d_y)}{\partial x}\right)^2.\Psi_m^e(x,y).\Psi_n^e(x,y)$$

$$R_{mn}^{k,yx} = 2.\sum_{e\in supp(mn)}\sum_{(x,y)\in e}\left(\frac{\partial I_1(x-d_x,y-d_y)}{\partial x}\right)\left(\frac{\partial I_1(x-d_x,y-d_y)}{\partial y}\right).\Psi_m^e(x,y).\Psi_n^e(x,y)$$

$$R_{mn}^{k,yx} = R_{mn}^{k,xy}$$

$$R_{mn}^{k,yy} = 2.\sum_{e\in supp(mn)}\sum_{(x,y)\in e}\left(\frac{\partial I_1(x-d_x,y-d_y)}{\partial y}\right)^2.\Psi_m^e(x,y).\Psi_n^e(x,y)$$

where m and n denote nodes of the meshing and take in turn the values $(1\ldots N)$ and where supp(nm)=supp(n) $\cap$ supp(m).

- $\alpha = \max_n\left(\left|\nabla_n E^k\right|.\left\|\Psi_n\right\|\right)$

where $\left\|\Psi_n\right\|$ is a functional norm of Ψ_n. The two most advantageous norms are :

$$\left\|\Psi_n\right\| = \sup_{(x,y)\in supp(n)}\left|\Psi_n(x,y)\right| = 1 \qquad \text{or}$$

$$\left\|\Psi_n\right\| = \sqrt{\frac{1}{\left|sup\,p(n)\right|}\sum_{(x,y)\in supp(n)}\left[\Psi_n(x,y)\right]^2}$$

|supp(n)| denotes the cardinal number of the discrete region sup p(n).

At the end of this optimisation phase, there are available N nodal vectors each relating to one node of the meshing.

According to a variant embodiment of the adaptive gradient, it is possible to consider using a decomposition technique known as an "LDL^t profile" in technical language so as to resolve the linear system (1) and accelerate the treatment. This technique is described in the work entitled "Matrix numerical analysis applied to the art of the engineer" by Théodore Lascaux, Volume 1, pp 295-299, published by Masson, 1986.

According to an advantageous characteristic of the invention, the meshing is next locally refined via the division of triangles when the variation between the image $\hat{I}_2$ and the image I_2 on these triangles is too high. This is why according to the step (c) of the method a variation E is calculated between the image $\hat{I}_2$ and the image I_2 for each triangle e. The variation E is defined as follows :

$$E = \sum_{(x,y)\in e} DFD^2(x,y)$$

with $DFD(x,y) = Y_2(x,y) - Y_1(x-d_x, y-d_y)$

Of course, so as to calculate this variation for each triangle, it is necessary to firstly have calculated the value of the movement vectors of all the points of the image I_2 by means of interpolation from the nodal vectors obtained at the end of step (b).

Then in accordance with step (d), the meshing is refined on a discrete fraction of all the triangles of the model. This fraction is determined according to a criterion relating to the variations E previously calculated in step (c). So as to carry out this refining, it is possible for example to classify the triangles of the model by a decreasing order of their variations E and subdividing the X first triangles of this classification into smaller triangles. X is a predetermined fraction of the number of finished elements in the model, for example half.

So as to locally refine the meshing, it is also possible to compare all the variations E calculated in step (c) with a threshold variation which depends on the size of the finished element in question and of subdividing into smaller finished elements the finished elements whose variations E are greater than the threshold variation.

The subdivision of a triangle e into four smaller triangles is shown on figure 2. The triangle e is defined by the three vertices P_1, P_2 and P_3 having for respective movement vectors $\vec{d}_1$, $\vec{d}_2$, $\vec{d}_3$. So as to subdivide it into four, three new nodes P_4, P_5, P_6 are defined in the middle of the three sides P_1P_3, P_1P_2, P_2P_3 of the triangle. Allocated to each of these three new nodes is a movement vector equal to the

average of the movement vectors of the two tops of the side to which it belongs, respectively $(\vec{d}_1+\vec{d}_3)/2$, $(\vec{d}_1+\vec{d}_2)/2$, $(\vec{d}2+\vec{d}_3)/2$. The adjacent triangles to the triangle e whose side is P_1P_2, P_2P_3 or P_1P_3 are then subdivided into two or three.

Thus a model of finished elements is obtained whose meshing has been locally refined. According to step (e), steps (b), (c) and (d) are repeated at the end of the preceding step (d). This succession of steps is then repeated until a stoppage criterion is satisfied. This stoppage criterion is for example a predetermined number of finished elements to be reached at the end of step (d).

It is also possible to stop the method when the variations E of all these finished elements of the model obtained at the end of the preceding step (c) are lower than a threshold variation.

According to an improved embodiment shown on figure 3, the steps (b) to (e) are carried out by depending on several resolution levels of images I_1 and I_2. The aim of this variant is to improve and accelerate the convergence of the calculations of the movement vectors. In order to achieve this, first of all for each pair of numerical images I_1 and I_2, a set of R images I_i^r is defined with a level of resolution r and luminance Y_i^r, r taking in turn the values (R-1, R-2,...0) and i the values (1, 2) and then steps (b) to (e) are carried out for each level of resolution r from the level of resolution r=R-1 to the level r=0. It is to be noted that the images I_1^0 and I_2^0 correspond to the numerical images I_1 and I_2.

In practice, the images I_j^r are obtained by the filtering of the image I_i using a linear low pass filter only allowing $1/2^r$ of the spectral band of the image in question in the directions x and y, that is a pulse response filter h_n^r having a pass-band $BP_r=[1/2^{r+1}, 1/2^{r+1})$ in the space of standardised frequencies [-1/2, 1/2]. The image I_i^r is defined by the following equation :

$$Y_i^r(x,y)=\sum_{u=-M}^{M}\sum_{v=-M}^{M}Y_i(x-u,y-v)h_u^r h_v^r$$

The filter used is for example an approximation of an ideal filter and its pulse response is defined as follows :

$$h_n^r = \frac{s_n^r}{S} \quad \text{with} \quad S = \sum_{n=-M}^{M} s_n^r \qquad -M \le n \le M$$

$$\text{and } s_n^r = 2B.\operatorname{sinc}(2\pi B_r n) = 2B\frac{\sin 2\pi B_r n}{2\pi B_r n}$$

$$B_r = \frac{1}{2^{r+1}}$$

where B and M are natural integers
$M = +\infty$ in the ideal case.

As indicated previously, this optimisation on several resolution levels is able to improve and accelerate the convergence of the calculations of the movement vectors. It is to be noted that the number of resolution levels R selected may differ from the number of successive refinings carried out on the meshing.

According to a preferred embodiment, compactness constraints are added on each triangle of the model so as to prevent the triangles from reversing.

The compactness of a triangle with vertices P_i, P_j and P_k is defined by the following equation :

$$C(P_i, P_j, P_k) = \frac{4 \times \pi \times S(P_i, P_j, P_k)}{P^2(P_i, P_j, P_k)}$$

with $C(P_i,P_j,P_k) \in]0,1[$; and $S(P_i,P_j,P_k)$ et $P(P_i,P_j,P_k)$ representing respectively the surface and perimeter of the triangle (P_i, P_j, P_k)

If the compactness of a triangle is prevented from tending towards zero, it is also prevented from reversing. This is why, so as to avoid the reversals of triangles, each triangle must verify the following constraint :

$$C\left(P_i + \vec{d}_{P_i}, P_j + \vec{d}_{P_j}, P_k + \vec{d}_{P_k}\right) \geq K \times C\left(P_i, P_j, P_k\right)$$

$$\Leftrightarrow K \times C\left(P_i, P_j, P_k\right) - C\left(P_i + \vec{d}_{P_i}, P_j + \vec{d}_{P_j}, P_k + \vec{d}_{P_k}\right) \leq 0$$

$$\Leftrightarrow g_e\left(\vec{D}\right) \leq 0 \ ; \quad e = \text{triangle}\left(P_i, P_j, P_k\right)$$

where K is a parameter fixing the authorised compactness variation and $\vec{D}$ is the column vector of the movement vectors of the nodes of the model.

According to the invention, this constraint is associated with each triangle at the time the movement vectors are optimised. The step for optimising the movement vectors amounts to a system of the type :

$$\begin{cases} \min_{\vec{D}} E\left(\vec{D}\right) \\ g_e\left(\vec{D}\right) \leq 0 \qquad \forall e \in I \\ \vec{D} \subset \Re^{2N} \end{cases}$$

where:

- $E(\vec{D})$ represents the variation between the image I_2 and the said image $\hat{I}_2$;
- g_e is a constraint related to the triangle e ;
- I is the set of the triangles of the meshing.

So as to resolve the optimisation problems under constraints, the so-called increased Lagrangian technique is used. This technique is described in the work entitled "Theories and algorithms" by Michel Minoux, Volume 1, pp 257-260, published by Dunod 1983. This technique combines two optimisation techniques : Lagrangian optimisation and the optimisation of external penalties.

According to this technique, resolving the preceding system amounts to resolving the system without constraints according to :

$$\min_{\vec{D}}\left(E\left(\vec{D}\right) + \sum_{e \in I} G\left(g_e\left(\vec{D}\right), \lambda_e, r_e\right)\right)$$

Where r_e is a penality element,

λ_e is a Lagrange multiplier

G is an increased Lagrangian determined by the equation :

$$G\left(g_e(\vec{D}),\lambda_e,r_e\right)=\begin{cases}\lambda_e g_e(\vec{D})+r_e g_e(\vec{D}) & \text{si } r_e>0 \text{ et } g_e(\vec{D})\geq 0\\ \lambda_e g_e(\vec{D}) & \text{si } r_e=0 \text{ et } g_e(\vec{D})\geq 0\\ 0 & \text{si } g_e(\vec{D})\leq 0\end{cases}$$

The constraints g_e have been previously linearised by the Taylor formula to the order 1 :

$$g_e(\vec{D})\approx g_e(P_i,P_j,P_k)+\sum_{P_l=P_i,P_j,P_k}\vec{d}_{P_l}\frac{\partial g_e}{\partial \vec{d}_{(P_i,P_j,P_k)}}(P_l)$$

The optimisation method is then the following :

- k=0 is initialised
- λ = 0 and r=0 are placed, $\lambda\in\Re^m$ and $r\in\Re^m$, and m denotes the number of triangles of the model
- then the minimum $\delta D_{k+1}(\lambda,r)$ is determined so that on iteration k+1,

$$-H.\delta D^{k+1}=\nabla E^k - C^t\gamma$$

where $\gamma^t=(\lambda,r)$, C^t is a matrix of $\Re^{2N}\times\Re^{2m}$

$C^t\gamma$ forms a matrix of the linearised constraints having for coefficients the following values :

$$C_{ij}=\begin{cases}0 & \text{si } g_e(\vec{D})\leq 0\\ \begin{cases}\partial_{\vec{d}_{P_l}} g_e(P_i,P_j,P_k) & \text{si } P_l=P_i,P_j \text{ ou } P_k\\ 0 & \text{sinon}\end{cases} & \text{sinon}\end{cases}$$

where e represents the triangle with the vertices P_i, P_j and P_k.

Then λ is updated by the Uzawa algorithm and r is increased. Then the preceding operation is repeated until all the constraints are verified before moving on to iteration k+2. It is to be noted that details of the Uzawa algorithm are given

in the work entitled "Theories and algorithms" by Michel Minoux, Volume 1, published by Dunod 1983.

According to a final embodiment, it is also possible to introduce constraints so as to avoid the flowing over of the meshing obtained after applying the movement vectors beyond the range of the image I_1. This embodiment consists of forcing the peripheral nodes of the meshing to remain on the edges of the image after applying movement vectors. In order to do this, the abscissae components δD^{k+1} for the peripheral nodes on the left and right edges of the image I_2 are cancelled on each iteration k. Similarly, the ordinate components δD^{k+1} for the upper and lower edges of the image I_2 are cancelled on each iteration k.

In the case of a coding application, associated with the meshing is a partially quaternary tree. On each subdivision of the meshing (step d), an additional level is added in the tree. Each level of the tree then represents a meshing level and each node of the tree represents a triangle of the corresponding meshing level. The binary train generated during the coding step is obtained via a reading of the tree level by level. In this binary train, the movement vectors associated with each node of the tree are advantageously coded differentially with respect to the movement vectors of their father node when the latter exists and are arranged level by level. The corresponding decoding step consists of regenerating this tree from the binary train received derived from the encoder.